METHOD AND APPARATUS FOR A WEB APPLICATION SERVER TO PROVIDE FOR WEB USER VALIDATION

CROSS REFERENCE TO CO-PENDING APPLICATIONS

U.S. Patent Application No. 09/164,759, filed October 1, 1998, and entitled, "A Common Gateway Which Allows Applets to Make Program Calls to OLTP Applications Executing on an Enterprise Server"; U.S. Patent Application No. 09/164,932, filed October 1, 1998, and entitled, "A Multi-Client User Customized DOM Gateway for an OLTP Enterprise Server Application"; U.S. Patent Application No. 09/164,908, filed October 1, 1998, and entitled, "An Automated Development System for Developing Applications that Interface with Both Distributed Component Object Model (DOM) and Enterprise Server Environments"; U.S. Patent Application No. 09/164,933, filed October 1, 1998, and entitled, "Providing a Modular Gateway Architecture Which Isolates Attributes of the Client and Server Systems into Independent Components"; U.S. Patent Application No. 09/164,862, filed Octob er 1, 1998, and entitled, "Making CGI Variables and Cookie Information Available to an OLTP System"; U.S. Patent Application No. 09/164,623, filed October 1, 1998, and entitled, "A Gateway for Dynamically Providing Web Site Status Information"; U.S. Patent Application No. 09/164,756, filed October 1, 1998, and entitled, "Development System for Automatically Enabling a Server Application to Execute with an XATMI-complaint transaction MGR :Managing Transactions within Multiple Environments"; U.S. Patent Application No. 09/189,053, filed November 9, 1998, and entitled, "Cool ICE Batch Interface"; U.S. Patent Application No. 09,189,381, filed November 9, 1998, and entitled, "Cool ICE Debug"; U.S. Patent Application No. 09/188,628, filed November 9, 1998, and entitled, "Cool ICE Workstation Directory/File Browser"; U.S. Patent Application No. 09/188,840, filed November 9, 1998, and entitled, "Cool ICE Icons"; U.S. Patent Application No. 09/188,738, filed November 9, 1998, and entitled, "Cool ICE Service Templates"; U.S.

Replacements for pages 1 & 2
Part of Amendment A, paper #3.

Patent Application No. 09/189,383, filed November 9, 1998, and entitled, "Automatic Footer Text on HTML Pages"; U.S. Patent Application No. 09/189,615, filed November 9, 1998, and entitled, "Availability Message"; U.S. Patent Application No. 09/189,612, filed November 9, 1998, and entitled, "Cool ICE System Settings"; U.S. Patent Application No. 09/188,807, filed November 9, 1998, and entitled, "Cool ICE Service Handler"; U.S. Patent Application No. 09/189,611, filed November 9, 1998, and entitled, "Server Side Variables""; U.S. Patent Application No. 09/188,629, filed November 9, 1998, and entitled, "Cool ICE data Wizard"; U.S. Patent Application No. 09,189,365, filed November 9, 1998, and entitled, "Cool ICE Table Profiling"; U.S. Patent Application No. 09/189,160, filed November 9, 1998, and entitled, "Cool ICE Database Profiling";U.S. Patent Application No. 09/188,649, filed November 9, 1998, and entitled, "Cool ICE Column Profiling";U.S. Patent Application No. 09/949,213, filed November 24, 1999, and entitled, "Cool ICE Method and Apparatus for a Web Application Server to Maintain Logon Security Mapped to a Gateway Through Server Based Session Objects"; U.S. Patent Application No. 09/448,169, filed November 24, 1999, and entitled, "Cool ICE Method and Apparatus for a Web Application Server to Upload Multiple Files and Invoke a Script to Use the Files in a Single Browser Request"; U.S. Patent Application No. 09/448,165, filed November 24, 1999, and entitled, "Cool ICE Method and Apparatus for a Web Application Server to Provide an Administration System Using a Dual Set of Tiered Groups of Objects"; and U.S. Patent Application No. 09/188,725, filed November 9, 1998, and entitles "Cool Ice State Management" are commonly assigned co-pending applications incorporated herein by reference.

or no access to sensitive data when the user terminal site is not particularly secure. These features can be effectively combined with physical security procedures to provide many specialized security profiles. Each individual service within an Active Server Page (ASP) may be validated. Other solutions must still transmit sign on over the network for each service. Even though such transmissions may be encrypted or sent over a secure connection, they can still be susceptible to being accessed and decrypted by malicious users. The present approach enhances granularity of security. The UserValidation service is used to convert site specific user validation data to a UserID and Password.

From the system perspective, rather than defining several levels of data classification, the different classes of users and user sites are managed by identifying a security profile as a portion of those service requests requiring access to secure data. Thus, the security profile accompanies the data/service to be accessed. The user simply need execute the sign on procedure which correlates to the access permitted. This permits certain levels of data to be accessed by one or more of the several classes of user.

In the preferred mode of practicing the present invention, a user signs on to the gateway with a generic login protocol providing access as an unsecured user. All users of the gateway sign on in a similar fashion. Should the user request access to a secure function or to secure data, the user validation, rather than the secure service, manages the security profiling technique. The service request for secure access results in the user validation requesting such additional logon information as is required to permit the desired access. In this way, the web browser request is associated with security attributes so that each web user transaction attaches to the database management system object using the security obtained from the Cool ICE session object.

Replacement for page 8
Part of Amendment A
Paper #3.

Fig. 4 is pictographic view of the system of Fig. 3 with particular detail showing the organization and operation of the Cool ICE system 62, which is resident in the web server (see also Fig. 3). In this view, the client accesses the data base management system within the enterprise via internet terminal 64 which is coupled to the web server 68 by world wide web path 66. Again, the internet terminal 64 is preferably an industry standard computer utilizing a commercially available web browser.

The basic request/response format of the Cool ICE system involves a "service" (defined in greater detail below) which is an object of the Cool ICE system. The service is a predefined operation or related sequence of operations which provide the client with a desired static or dynamic result. The services are categorized by the language in which they were developed. Whereas all services are developed with client-side scripting which is compatible with internet terminal 64 (e.g., HTML), the server-side scripting defines the service category. Native services utilize Cool ICE script for all server-side scripting. On the other hand, open services may have server-side scripting in a variety of common commercial languages including Jscript, VBScript, ActiveX controls, and HTML. Because native services are developed in the Cool ICE script (run) language, greater development flexibility and variety are available with this technique.

Web server 68 provides processor 70 for ASP's which have been developed as open services 72 and a Default ASP 73 for invoking native services. After the appropriate decoding within a native or open service, a call to the necessary Cool ICE object 74 is initiated as shown. The selected service is processed by the Cool ICE engine 76.

Repository 80 is a storage resource for long term storage of the Cool ICE service scripts and short term storage of the state of a particular service. Further details concerning repository 80 may be found by consulting U.S. Patent Application Serial No. 09/448,165, filed November 24, 1999. The preferred mode of the present invention, the service scripts stored in repository 80 are typically very similar to mapper runs as described above. For a more detailed description of

Replacements for pages 19-24
Part of Amendment A
paper no. 3

MAPPER runs, MAPPER User Manual is available from Unisys Corporation and incorporated herein by reference.

Cool ICE engine 76 sequences these previously stored command statements and can use them to communicate via network 84 with other data base management system(s) (e.g., MAPPER) resident on enterprise server 86 and/or departmental server 88. The storage capability of repository 80 is utilized by Cool ICE engine 76 to store the state and intermediate products of each service until the processing sequence has been completed. Following completion, Cool ICE engine 76 retrieves the intermediate products from repository 80 and formats the output response to the client, which is transferred to internet terminal 64 via web server 68 and world wide web path 66.

Cool ICE Administrator 82 is available for coordination of the operation of Cool ICE system 62 and thus can resolve conflicts, set run-time priorities, deal with security issues, and serve as a developmental resource. Graphing engine 78 is available to efficiently provide graphical representations of data to be a part of the response of a service. This tends to be a particularly useful utility, because many of the existing data base management systems have relatively sparse resources for graphical presentation of data.

The combination of Cool ICE object 74, Cool ICE engine 76, and repository 80 permits a rather simplistic service request from internet terminal 64 in dialog format to initiate a rather complex series of data base management system functions. In doing so, Cool ICE engine 76 emulates an intranet user of the data base management system(s) resident on enterprise server 86 and/or departmental server 88. This emulation is only made possible, because repository 80 stores sequences of command language statements (i.e., the logic of the service request) and intermediate products (i.e., the state of the service request). It is these functions which are not available in ordinary dialog on the world wide web and are therefore not even defined in that environment.

Fig. 5 is a schematic diagram 90 of the software components of the Cool ICE system and the software components to which it interfaces in the preferred mode of the present invention. The client user of the Cool ICE system interfaces directly with web browser 92 which is resident on internet terminal 64 (see also Fig. 4). Web browser 92 is a commercially available browser. The only special requirement of web browser 92 is that it be capable of supporting frames.

Web browser 92 communicates with web server software 96 via internet standard protocol using HTML language using world wide web path 94. Web server software 96 is also commercially available software, which is, of course, appropriate for to the web server host hardware configuration. In the preferred mode of the present invention, web server software 96 is hosted on Windows IIS- based server available from Microsoft Corporation..

Cool ICE system software 98 consists of Cool ICE Object (the gateway) 100, Cool ICE service handler 102, Cool ICE administration 104, Cool ICE repository 106, and Cool ICE Scripting Engine 108. It is these five software modules which establish and maintain an interface to web server software 96 using COM interfaces and interface to Cool ICE's internal and external data base management systems.

Cool ICE object 100 is the interface between standard, commercially available, web server software 96 and the internal Cool ICE system scripting engine with its language and logic facility. As such, Cool ICE object 100 translates the dialog format, incoming HTML service request into internal Cool ICE requests for service. Intrinsic in this translation is a determination of the service category (see also Fig. 4) -- that is whether the service request is a native service (i.e., with a default Cool ICE server-side scripting) or an open service (i.e., with server-side scripting in another commercial language using the Cool ICE object 100).

The service request, received from Cool ICE object 100, is utilized by Cool ICE service handler 102 to request the corresponding service action script from Cool ICE repository 106 and

to open temporary state storage using Cool ICE repository 106. Cool ICE service handler 102 sequences through the service input variables of the object received from Cool ICE object 100 and transfers each to Cool ICE repository 106 for temporary storage until completion of the service request. Cool ICE service handler 102 retrieves the intermediate products from Cool ICE repository 106 upon completion of the service request and formulates the Cool ICE response for transfer to browser 92 via web server software 96 and world wide web path 94.

Cool ICE administration 104 implements automatic and manual control of the process. It provides for record keeping, for resolution of certain security issues, and for development of further Cool ICE objects. Interconnect 110 and interconnect 112 are software interface modules for communicating over the enterprise network (see also Fig. 4). These modules are dependent upon the remaining proprietary hardware and software elements coupled to the enterprise network system. In the preferred mode of the present invention, these are commercially available from Unisys Corporation.

Fig. 6 is a schematic diagram 116 showing the processing of a service request by the Cool ICE system. Screen 118 is the view as seen by the client or user at an internet terminal (see also Fig. 4). This screen is produced by the commercially available browser 120 selected by the user. Any such industry standard browser is suitable, if it has the capability to handle frames. The language of screen 118 is HTML 124. Hyperlinks 126 is used in locating the URL of the Cool ICE resident server. The components of the URL are as follows. In many instances, this will simply be the internet access provider of the internet terminal, as when the internet terminal is owned by the enterprise and the user is an employee. However, when the user is not an employee and the internet terminal is not necessarily owned by the enterprise, it becomes more likely that hyperlinks 126 identifies a remotely located server.

Icon 122 is a means of expressly identifying a particular service request. Such use of an icon is deemed to be unique. Additional detail concerning this use of an icon is available in U.S. Patent Application Serial No. 09/448,169, filed November 24,1999. Window area 128 provides for the entry of any necessary or helpful input parameters. Not shown are possible prompts for entry of this data, which may be defined at the time of service request development. Submit button provides the user with a convenient means to transmit the service request to the web server in which the Cool ICE system is resident.

Upon "clicking on" submit button 130, screen 118 is transmitted to web server 136 via world wide web path 132. As discussed above, world wide web path 132 may be a telephonic dial-up of web server 136 or it might be a long and complex path along the internet if web server 136 is remote from the originating internet terminal. Web server 136 is the software which

Fig. 10 is a detailed diagram 300 showing operation of the security system during the honoring of a service request. The user, operating industry compatible, personalized computer, workstation 302, formats a service requests via commercially available web browser 304. In the preferred mode of the present invention, this is accomplished by then making a call to the Cool ICE system. The user simply requests access to the Cool ICE home page by transferring web browser 304 to the URL of Cool ICE system. After the Cool ICE home page has been accessed, one of the buttons is clicked requesting a previously defined service request. For additional detail on the service request development process, see above and U.S. Patent Application Serial No. 09/449,213, filed November 24, 1999.

The service request is transferred to web server 314 via world wide web path 306. The service request is received by Cool ICE object 322 and translated for use within the Cool ICE system. The request is referred to the Cool ICE Engine Interface 331 via path 324. In the preferred mode of practicing the present invention, the Cool ICE Engine Interface 331 is equivalent to the MAPPER data base management system. The service request is passed to Cool ICE Service Handler 332 for retrieval of the command language script which describes the activities required of the data base management system to respond to the service request.

Cool ICE Service Handler 332 makes an access request of Cool ICE service portion 340 of repository 342 via path 338. It is within Cool ICE service portion 340 of repository 342 that the command language script corresponding to the service request is stored. The command language script is obtained and transferred via path 336 to service handler 332 for execution. Along with the command language script, a security profile, if any, is stored for the service request. As explained in U.S. Patent Application Serial No. 09/188,549, filed November 9, 1998, the security profile, if required, is added to the command language script file at the time of service request development by the service request developer. This security profile identifies

Replacements for pages 33-34
Part of Amendment A
Paper no. 3

which of the potential service requesters may actually be provided with a complete response. The security profile, if any, is similarly transferred to service handler 332 via path 336.

If no security profile has been identified for the service request, service handler 332 allows the execution of the command language script received via path 336 through access of remote database 316 via paths 318 and 320, as required. The response is transferred to Cool ICE object 322 via path 328 for conversion and transfer to workstation 302 via world wide web path 310.

However, if a security profile has been identified for the service request, service handler 322 requests the user to provide a user-id via path 330, Cool ICE object 322, and world wide web path 312. Service handler 332 awaits a response via world wide web path 308, Cool ICE object 322, and path 326. Service handler 332 compares the user-id received to the security profile stored with the command language script. If the user matches the security profile, access is granted and service handler 322 proceeds as described above. If the user does not match with the stored security profile, the service request is not executed and the user is notified via an appropriate message.

INS. E17